TECHNICAL REPORT

ISO/IEC TR 29181-3

First edition
2013-04-15

Information technology — Future Network — Problem statement and requirements —

Part 3:
Switching and routing

Technologies de l'information — Réseaux du futur — Énoncé du problème et exigences —

Partie 3: Commutation et routage

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide to publish a Technical Report. A Technical Report is entirely informative in nature and shall be subject to review every five years in the same manner as an International Standard.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 29181-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*.

ISO/IEC TR 29181 consists of the following parts, under the general title *Information technology — Future Network — Problem statement and requirements*:

⎯ *Part 1: Overall aspects*

⎯ *Part 3: Switching and routing*

⎯ *Part 4: Mobility*

⎯ *Part 6: Media transport*

⎯ *Part 7: Service composition*

The following parts are under preparation:

⎯ *Part 2: Naming and addressing*

⎯ *Part 5: Security*

Introduction

ISO/IEC TR 29181-1 describes the definition, general concept, problems and requirements for the Future Network (FN). The other parts of ISO/IEC TR 29181 provide details of various components of the technology.

This part of ISO/IEC TR 29181 examines the requirements for carrying data over digital networks, and identifies those that are not satisfied by the current Internet.

It also notes some expected characteristics of new systems that are better able to satisfy the requirements, and specifies a model which supports both the existing system and the new systems. This will enable a migration to the new systems; it is also intended to make networks of all sizes easier to manage.

Information technology — Future Network — Problem statement and requirements —

Part 3:
Switching and routing

1　Scope

This part of ISO/IEC TR 29181 contains the problem statement and requirements for switching and routing in the Future Network, in particular:

a)　description of the requirements for carrying data over digital networks;

b)　description of the ways in which these requirements are not satisfied by current networks;

c)　functional architecture for switching and routing in the Future Network; and

d)　requirements for control plane information flows for finding, setting up, and tearing down routes.

The requirements in (d) include support for both current ("legacy") and future ("new") switching technologies, to aid the transition between them.

NOTE　　A distinction is made between "data", which is simply a string of bytes, and "content", for which an interpretation, for instance as text, sounds, or still or moving images, is defined. Content is addressed in ISO/IEC TR 29181-6.

2　Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC TR 29181-1, *Information technology — Future Network — Problem statement and requirements — Part 1: Overall aspects*

3　Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC TR 29181-1 and the following apply.

3.1
data unit
sequence of octets which is conveyed across the network as a single unit

3.2
flow
sequence of data units

3.3
asynchronous flow
flow consisting of data units for which the time of arrival at the destination is unimportant

3.4
synchronous flow
flow for which the network delay experienced by data units is required to be within specified limits, the size of the units, and also the number of units to be transmitted per unit time, being either fixed, or variable with a defined upper limit

3.5
connectionless data unit
data unit which is not part of a flow

3.6
network delay
time from submission of a data unit to the network by the sender to its delivery to the recipient

NOTE   The exact events which constitute submission and delivery are not specified in this document, as they depend on internal details of the equipment.

3.7
network element
piece of equipment which takes part in the process of conveying data, such as a switch, gateway, or interface

3.8
FN-aware network element
network element which implements a protocol which will be standardised, based on the requirements in clauses 9 and 10

3.9
legacy network
network composed of network elements which are not FN-aware

3.10
end equipment
equipment that is connected to the network and produces or consumes data units

3.11
identifier
value that identifies a subnetwork, network element, service, or piece of content

3.12
locator
value that identifies a subset of the network which is the scope of an identifier or in which the object identified by an identifier is to be found

3.13
address
identifier, or locator together with an address whose scope is defined by the locator

3.14
label
information in the encapsulation of a data unit which defines, to the network element which receives the data unit, how it is to be routed

EXAMPLE 1   IP address and port number

EXAMPLE 2   MPLS label

EXAMPLE 3   ISDN channel number

3.15
encapsulation
additional octets or other symbols associated with a data unit which serve to delimit it or to identify aspects of the service it should receive

3.16
packet
data unit together with its associated encapsulation

4   Abbreviations

Numbers in square brackets identify references in the Bibliography. Numbers prefixed by RFC identify IETF "Request For Comments" documents.

ADSL   Asymmetric Digital Subscriber Line
ARP   Address Resolution Protocol (RFC 826)
AoIP   Audio over Internet Protocol
ATM   Asynchronous Transfer Mode
AVB   Audio Video Bridging [1]
CO   Connection Oriented
CL   ConnectionLess
DAB   Digital Audio Broadcasting
DHCP   Dynamic Host Configuration Protocol (RFC 2131)
DNS   Domain Name System (RFC 1034, 1035)
DSL   Digital Subscriber Line
DWDM   Dense Wavelength Division Multiplexing
EUI   Extended Unique Identifier
FEC   Forward Error Correction
FM   Frequency Modulation
GPS   Global Positioning System
HTTP   HyperText Transfer Protocol (RFC 2616)
IANA   Internet Assigned Numbers Authority
IEEE   Institute of Electrical and Electronics Engineers
IETF   Internet Engineering Task Force
IP   Internet Protocol
IPv4   Internet Protocol version 4 (RFC 791)
IPv6   Internet Protocol version 6 (RFC 2460)
ISDN   Integrated Services Digital Network [2]
ISP   Internet Service Provider
MAC   Media Access Control
MPLS   MultiProtocol Label Switching (RFC 3031)
MTU   Maximum Transmission Unit size
NAT   Network Address Translation
OSI   Open Systems Interconnection
PC   Personal Computer
PCM   Pulse Code Modulation
QoS   Quality of Service
RSVP   resource ReSerVation Protocol (RFC 2205)

SDH   Synchronous Digital Hierarchy [3]

SDP   Session Description Protocol (RFC 4566)

SIP   Session Initiation Protocol (RFC 3261)

SNAP   SubNetwork Access Protocol

SNMP   Simple Network Management Protocol (RFC 1157)

STL   Studio-Transmitter Link

STUN   Session Traversal Utilities for NAT (RFC 5389)

TCP   Transmission Control Protocol (RFC 792)

UDP   User Datagram Protocol (RFC 768)

UHF   Ultra High Frequency

URL   Universal Resource Locator

VCI   Virtual Channel Identifier

VPI   Virtual Path Identifier

5   Background

5.1   Main features

New switching technologies are expected to support virtual circuits, in contrast to the connectionless paradigm of the current Internet Protocol: see 5.2.

The control protocols outlined in this document allow an application to negotiate with the network, and with the remote application, to achieve the communication of its data in the most appropriate way. They do not constrain the system to use any particular method of delivering the data to its destination, and thus support both existing and new switching technologies. They do, however, require the network to report explicitly (to the application) parameters defining the service it will receive; this allows the application to take advantage of the facilities offered by the new networks if available, and to make provision for the deficiencies of existing technologies otherwise.

The Internet Protocol world is often depicted as an hour-glass, with many applications at the top, many network technologies at the bottom, and IP as the narrow waist through which everything must pass. FN has a similar structure, but there are some important differences in the nature of the narrow waist:

- there are two kinds of interaction between the application and the network, which we describe as control plane and data plane;

- control plane messages usually refer to a "flow" (a sequence of data units) rather than to a single data unit, so are able to include the QoS parameters needed for live media, also other session-related information such as security checks and per-call billing; and

- the data plane carries data units whose encapsulation depends only on the local switching technology.

Thus different kinds of switching technology can be supported, even those (such as DWDM) that do not route according to information in packet headers. Where packet switching is used, the routing information in the packet does not need to include the kind of globally unique addresses that are required for IP.

5.2   Rationale for a new paradigm

5.2.1   Appropriateness for today's traffic

Today's Internet mainly carries two kinds of traffic: static objects such as files and web pages, and continuous media such as audio and video. The connectionless service provided by Internet Protocol is transmission of individual packets from one interface to another; it is a "best effort" service, aiming to deliver each packet as soon as possible but not giving any guarantees as to when (or even whether) each packet will be delivered.

A user downloading a file simply requires a copy of the file to be received by a particular piece of equipment (such as a PC) as soon as possible after it is requested, so a "best effort" service such as that provided by IP is appropriate. However, the equipment may have several different interfaces to the network, and the user should not need to consider through which interface the data should be received, nor the location of the server on which it is hosted.

In the case of continuous media, the source produces data at regular intervals and the destination consumes it at regular intervals. The time between the source sending some data and the destination consuming it must be long enough for the slowest packet to arrive. (The destination must hold data packets that arrive more quickly in a buffer.) In connectionless packet networks, this time interval includes an unpredictable amount of queuing time and is thus not well-defined. For some applications, such as streaming recorded material, this is of minor importance, but for others, such as videoconferencing, it degrades the user experience and in some cases, for instance some telemedicine and industrial control applications, delays will compromise safety.

To deliver a service appropriate for continuous media, the network needs to configure resources to be ready to pass on the data with a minimum of delay; a negotiation is therefore required between the application and the network before transmission begins. Various measures have been introduced to improve the service experienced by continuous media on the current Internet, but this traffic still experiences dropped packets and unnecessarily long delays. Moreover, the addition of new protocols and procedures increases the complexity of the system, decreasing its reliability and making it more difficult to manage: see reference [4] in the Bibliography.

5.2.2   Addressing and scalability

Use of Internet Protocol also requires every packet to carry the IP addresses of sender and destination. These addresses have global scope, and must therefore be large enough to uniquely identify every endpoint; as the number of endpoints increases (a process that will accelerate as the Internet of Things develops), addresses need to become larger, for instance changing from 32-bit IPv4 addresses to 128-bit IPv6 addresses. A system with a fixed address format in which addresses have global scope cannot scale beyond a certain size; for instance a system using IPv4 cannot have more than 4 billion endpoints.

However, a switch or router simply needs to forward the packet to the correct neighbour with the correct level of service, and locally-significant forms of identification have the potential to allow this task to be performed more efficiently and more reliably. Moreover, they do not impose any limitations on the total size of the network because addresses in one area can be re-used in other areas. Network address translation has retro-fitted this feature to IPv4, but caused other problems which required the introduction of further protocols such as STUN.

FN makes possible a migration from IP to improved switching technologies in which the information used for local routing is separate from global addressing.

5.2.3   Security

Another area of concern is security. In a connectionless packet-switched network, all the information needed to route a packet has to be included in the packet header. This limits the amount of checking that can be done without unreasonably increasing the packet size, or the amount of processing required per packet at each switching point, or both. A system in which the routing information in a packet refers, indirectly, to a route that was set up as a result of a negotiation which can include much more thorough checking of identity etc, has the potential to be much more secure. It can also enhance privacy by allowing a client to set up a session with a server without disclosing the client's address to the server.

5.2.4   Complexity of routing hardware

Connectionless packet switching requires each switch or router to read and interpret addressing information in each packet, in order to discover on which output to forward it and what service (e.g. higher- or lower-priority queue) it should experience. This information is usually too long to be used as a memory address for direct lookup, for instance an IPv6 address is 128 bits and the information necessary to identify a flow in an IPv4 packet is over 100 bits, so the more complex and power-hungry associative or content-addressable memory must be used. Other techniques, where much of the work is done once (when the call is set up) rather than for

every packet, allow switching equipment (and particularly large switches) to be simpler and less power-hungry; see, for instance, 4.1 of reference [6]. With increasing traffic on the Internet, and increasing concern over energy use and its contribution to climate change, efficiency of switching equipment will become more important over the coming decade.

5.3   Migration

The FN model allows a network to be seen as a "black box", so it is only the edge devices that need to be FN-aware. If the network supports QoS, its native QoS mechanisms can be used and the resulting QoS parameters reported to the application. Otherwise, the application will be informed that only a "best effort" service is possible, with an indication of the performance expected (for instance, whether the network is the public Internet or a small Ethernet network). This allows piecemeal migration in both private and public networks.

For instance, on an AVB network [1] the resource reservation mechanisms of IEEE 802.1Qat and Qav can be used to deliver defined QoS. The edge devices need to support the FN protocols, but within the network only the AVB protocols need to be supported.

An application on an FN-aware terminal on the AVB network can ask to receive video from a server identified by a URL; if the server is on the AVB network, IEEE 1722 can be used for transport and there is no need to use IP at all. If the server is on the Internet, there are two possibilities: either the application is told it has to use IP over the Ethernet network, or the gateway between the AVB network and the Internet implements the IP protocols.

A public network which supports non-IP routing may tunnel IP packets through its asynchronous service, and it may convert between IP and FN protocols in gateway devices. It may intercept protocols such as SIP and set up synchronous flows for them.

On the link between two networks, for instance an ADSL link between a private network and an ISP, if the devices at both ends of the link are FN-aware the FN protocols can be used. Otherwise, legacy protocols must continue to be used.

5.4   Structure of this document

Clause 6 lists requirements for applications that transmit information across digital networks, and clause 7 lists ways in which the current technology fails to meet those requirements.

Clause 8 examines the functionality of switching and routing equipment. Clauses 9 and 10 list requirements for the communication between network elements, and clause 11 outlines the requirements for the process of exchanging information between network elements that allows the route to any given destination to be established.

6   Requirements

6.1   Introduction

The basic service provided by the network is to convey data units between end equipment.

This clause reviews the requirements which applications that transmit various kinds of data across digital networks have for this service.

Many different kinds of application will use this service to convey information encoded as octet strings, and the network should not attempt to interpret the data units it conveys. It should not deliberately alter them in any way, although transmission errors may occasionally occur (see 6.4.2). Many different kinds of network element will participate in the conveying of an application's data unit, and the application should not make assumptions about their characteristics.

In most cases, the application will need to send a sequence of data units, either because the data to be sent is larger than the maximum size of a data unit (as with file transfer) or because it is being produced and/or consumed by a continuous process (as with audio and video). Such a sequence is referred to in this document as a "flow".

Each sub-clause below describes a group of requirements that have been identified for the carriage of data units. Mostly, these are expressed as requirements on flows rather than on individual data units.

6.2 Classes of flow

6.2.1 Minimising complexity

- The number of different options should be kept to a minimum.

This makes the task of implementing and managing networks as simple as possible. Thus, only two classes of flow are described in this document; it would be possible to identify additional classes, but the improvement in efficiency achieved by tailoring the service more closely to the application requirements would be outweighed by the increase in complexity of the system.

For example, no distinction is made between constant bit rate and variable bit rate media flows. We assume that in the latter case the application will request enough capacity to support its maximum bit rate; when it is sending at a lower rate, for some switching technologies the network will be able to use the remaining bandwidth for best-effort traffic, while for others the "wasted" bandwidth is less than the additional overheads of alternative mechanisms.

6.2.2 Synchronous flows

6.2.2.1 Network delay

- The application should be able to negotiate with the network to achieve the best balance between the delay the application can tolerate and the delay the network can achieve.

For many applications the maximum network delay is important, but the acceptable limits vary from a few milliseconds to a few seconds. Also, the recipient needs to have enough buffer space to store data that arrives with a minimum delay until it is needed, so delay variation can also be important.

For some applications, particularly safety-critical applications that involve control loops (whether or not humans are included in the loop), the maximum delay must not be exceeded; these applications require absolute guarantees from the network. The maximum delay value depends on the parameters of the control loop.

For other applications, particularly those involving conversations between humans, there is no hard-and-fast limit to the delay, but as the delay increases communication becomes increasingly difficult. Delays of a few tens of ms are acceptable in conversation, but if the round trip delay is more than 150 ms it is difficult to avoid both parties speaking at once and the conversation does not flow naturally; thus for conversational services the end-to-end time in each direction should be less than 75ms including encoding and decoding delays. In the case of audio- and video-conferencing, the signals may go via a central unit, in which case the round trip will involve four transits across the network so the end-to-end time for each transit needs to be less than 37.5 ms. If the time to encode the signals in digital form, and decode them at the receiving end, is significant, the budget for network transmission is reduced still further.

NOTE   Audio analogue-to-digital and digital-to-analogue converters typically take 1ms each. Encoding in a compressed form can take tens or hundreds of milliseconds.

The most demanding requirements are probably where musicians' sound is returned to them through in-ear monitors; a total round trip time (via a mixing desk, including two transits across the network as well as processing delays) of as little as 1 ms is perceptible by some performers, and most have difficulty in

performing above about 40 ms. (Sensitivity is different for different instruments, with vocalists being particularly sensitive; for details see [7].)

In unidirectional transmission, longer delays are tolerable but there will be a requirement to synchronise content delivered by different routes, for instance when handing over between devices (see 6.6) or when combining a programme received over the air with additional material received over the Internet.

6.2.2.2 Throughput

• A wide range of data unit sizes and transmission rates should be supported.

At one extreme, a telemetry application may transmit a data unit consisting of a small number of bytes once a minute. At the other extreme, uncompressed high-definition video requires multiple gigabits per second.

An example from legacy networks is 64 kb/s ISDN, where a single-byte data unit is transmitted every 125 µs, synchronised to the network.

For some switching technologies, reservations will be for data units to be transmitted at a specified regular interval, with the data units having a specified maximum size; such technologies are likely to be able to achieve the lowest latencies by synchronising incoming and outgoing streams at a switch. For others, they may be for a specified total number of bits per unit time; this figure would have to include encapsulation, which is technology-specific, so should still be expressed as a number of data units of a specified size.

6.2.2.3 Data unit size

• The network should inform the application of the maximum data unit size and maximum per-data-unit overhead for the route a flow will follow, so that it can choose an appropriate encapsulation for the data.

For most media flows, there will be a "natural" data unit size; for instance, in the case of linear PCM audio it is one sample per channel. If this is more than the maximum data unit size, the application will need to split the data into several units, for instance if there are 250 channels and up to 128 will fit in a maximum sized data unit, it should send channels 1-125 in one data unit and 126-250 in another.

In the case of compressed audio, the natural size of a data unit will be one frame of compressed data, typically 20 ms of audio.

If the natural data unit size is small compared with the reported per-data-unit overhead, the application should use aggregation to create larger data units. In the case of linear PCM audio, this means putting more than one sample per channel into each data unit, though this will increase the delay. (For example, if the sampling rate is 48 kHz and a data unit contains 48 samples per channel, an extra 1 ms of delay is introduced: the first sample in a data unit waits for 1ms before it is transmitted, and the last sample waits an extra 1ms at the receiving end before it is consumed.) The appropriate limits for overheads will depend on circumstances; overheads of several hundred percent are likely to be acceptable when sending audio on a gigabit Ethernet local network, whereas a public network may reject, or charge a premium for, calls which send a large number of small packets.

With some technologies, data units may be framed in such a way that there is no upper bound to their size; the format of the message reporting the maximum size should support this case.

NOTE   The application is responsible for sending data units of the right size; the network does not do any fragmentation or reassembly.

6.2.2.4 Routing

• FN should not place any restrictions on the technology that can be used for routing synchronous data units. In particular, it should not require a data unit, or its encapsulation, to include a globally-significant address: see 6.7.

There are a number of ways in which the flow to which a data unit belongs may be identified, for instance by a local "handle" value such as an ATM VCI, or by other information from the encapsulation such as its position in an ISDN frame or the wavelength of its carrier.

Synchronous flows should be able to pass seamlessly along a route which includes several different routing technologies.

6.2.2.5   Directionality

• The synchronous service should be unidirectional.

In many cases a synchronous flow is inherently unidirectional; either there is no return path or the return path carries a different kind of data. A radio or television broadcast is unidirectional, likewise video-on-demand (except for very occasional control messages). An outside broadcast sends programme-quality material to the studio, but the return path may be a lower-quality channel, or (for live broadcasts) the off-air signal may be used (see A.1).

Where bidirectional communication is required, as with a telephone call, two flows can be set up; typically the network will set both flows up at the same time, and by the same route, so the extra effort compared with a bidirectional service is minimal.

6.2.2.6   Multicasting

• One-to-many transmission, with the network copying the data as necessary, is required, and should support very large numbers of recipients for a single flow.

There are many applications in which it is useful for the network to copy the data to more than one recipient. For instance when a TV station covers an event such as a football match there will be many people wanting to watch it simultaneously.

One-to-one transmission is simply a special case of one-to-many in which there is only one recipient. Defining all flows as potentially one-to-many also simplifies handover when mobile devices move to a different attachment point, or when a user switches to a different device, by using the sequence: set up the new route; switch to using data arriving by the new route; tear down the old route.

Where an application requires a many-to-many service, this can be constructed from a number of one-to-many flows, in the same way that a bidirectional flow is constructed from two unidirectional flows. In practice, a different configuration will often be more efficient. For instance, in an audio conference, instead of sending all the audio streams to every participant, they will typically be sent to a central point which mixes them and then multicasts the result.

6.2.3   Asynchronous flows

6.2.3.1   General

Asynchronous flows experience a "best effort" service; the network makes no promises as to when, or even whether, each data unit will arrive at its destination. Some technologies ensure that data units that are part of the same flow arrive in the order in which they were transmitted; with others, data units within a flow are able to overtake each other.

Network delay and throughput cannot therefore be guaranteed for asynchronous flows. However, there should be provision for the application to give the network an estimate of the amount of data that will be sent, and for the network to give the application an estimate of the throughput the route will support.

As with synchronous flows, the network should inform the application of the maximum data unit size for the route a flow will follow; the maximum per-data-unit overhead is less important, but should also be reported if available. This means the network does not need to be able to fragment and reassemble data units; see also 7.2.2.

6.2.3.2   Routing

- The network should not place any restrictions on the technology that can be used for routing asynchronous data units. However, connection-oriented routing is preferred.

Asynchronous routing can be connection-oriented (CO), in which a route is first set up and data units are then sent along that route, or connectionless (CL), in which each data unit carries with it all the information needed to route it. Thus CL data units need more information in the header, so the header is larger, and requires more processing. Downloading a film requires transferring of the order of 1 million data packets if the size of each is 1500 bytes, so the amount of additional processing is significant. With protocols such as TCP, a large proportion of packets are acknowledgements and thus have a very small payload. Again, overheads will be less if the packet headers can be kept small.

For applications where only one or two data units are sent to a particular destination, CL routing might be thought to have the advantage that the separate set-up and tear-down mechanisms are not required. However, most of their functionality still has to be implemented; for instance, when a packet for a previously unknown destination arrives at a switch the same routing decisions need to be made as for a CO set-up. Also, it is easier to detect and eliminate loops in a route during connection-oriented call set-up than when routing connectionless packets.

In practice, typical Internet transactions (such as those used when "web surfing" via HTTP) involve multiple packets over TCP, which is a session-oriented byte-stream protocol and thus well suited to running over a CO network. Indeed, initiating a TCP session with a new destination, including discovering the route the packets must follow, requires the same processes as setting up a connection. However, in practice most of the route to the destination will already be cached in the network, so the set-up process will be somewhat quicker.

- CO networks should provide a mechanism for fast set-up of such calls, so that setting up a session with a server is at least as fast as on a CL network; such a mechanism is described in 9.2.

6.2.3.3   Directionality

- Asynchronous flows should be unidirectional, though they will often be used in pairs, one in each direction.

Whereas data on a synchronous flow will always arrive at its destination provided there are no equipment failures (or, for wireless networks, transmission problems) and the traffic contract is not exceeded, data units on asynchronous flows are always liable to be lost. Thus applications which use a "best-effort" service will normally use protocols in which there is a positive indication that each data unit has arrived successfully, either simple acknowledgements (as with TCP) or higher-layer replies (as, for instance, with SNMP requests), so it is appropriate for asynchronous flows to be bidirectional.

However, in the mechanism described in 9.2 a single data-plane path is used for data units from multiple clients, with each client having its own return path. For most switching technologies, even where communication is one-to-one the two directions will be set up separately within the routing tables.

6.2.3.4   Multicasting

- Multicasting is not a requirement for asynchronous flows.

Most of the traffic for which multicasting is useful will be carried on synchronous flows. Moreover, when multicasting over a best-effort service, a positive acknowledgement from each recipient that the data unit has arrived will normally be required, which does not scale well to large numbers of recipients.

Also, routing is simpler if each data unit only needs to be put in one queue. (If synchronous data units are routed synchronously, they can be copied to different outputs without adding them to multiple queues. Thus the multicasting of 6.2.2.6 is easier to support.)

One scenario where a best-effort multicast without a return path could be useful is in reporting status using a regime in which immediate reports are sent when there is any change, with periodic repetition of the status

when there is no change, for the benefit of those joining the multicast and those that have missed a message. However, in most cases this can equally-well be done using a synchronous service.

Where files need to be distributed to a large number of users, it would be better to use local caching. This also better supports applications such as video-on-demand, where a large number of users want to download the same content but not at exactly the same time.

6.3 Addressing

- The network should support the use of any current or future address type. Addressing should not be constrained by the switching technology, particularly considering that a flow may pass across several subnetworks using different technologies.

The addresses used by applications should therefore be independent of the mechanism used by the switching technology to route data units.

An address serves to indicate the destination for transmitted data units, and the source of received data units. It may contain several components, each of which may function as a locator, which specifies a subset of the network, or an identifier, which specifies an individual piece of equipment or content.

Addresses in current use include URLs, IPv4 and IPv6 addresses, EUI-64, IEEE 802 MAC addresses, and E.164 telephone numbers. Future applications may find other kinds of addressing useful, for instance GPS co-ordinates.

Locators are not needed in networks that are small enough that each network element can know the location and capabilities of all the other network elements. In larger networks they serve to limit the complexity of the process of finding a route. An address may include a hierarchy of locators, for instance an E.164 telephone number includes a country code and an area code.

Applications need to be able to address many different kinds of entity, for instance networking equipment, user terminals, points of attachment to the network, and pieces of content. These entities may be addressed in different ways, for instance by an identifier (which would address it uniquely, except for content where there may be multiple copies) or as the provider of a particular service (of which there may be many, but the application does not care which is chosen).

6.4 Security and accountability

6.4.1 Privacy, trust and traceability

- The network should be designed to keep private the contents, and even the existence, of calls.

A significant part of the traffic carried by telecommunications networks is of a private nature, for instance commercial discussions which must not be revealed to competitors. Encryption can provide additional protection for data known to be particularly sensitive, such as credit card details.

- The network should be able to verify, or at least assign a measure of confidence to, information such as the identity of a calling party and the route taken by a call.

This helps to identify calls that come from, or are routed through, less trustworthy parts of the network.

6.4.2 Corruption of data units

- The network should ensure that almost all the data units it delivers are an exact copy of the data unit submitted to it by the sender, but it should not attempt to detect bit errors in individual data units.

The effect of bit errors in the data can be very different for different applications; see 9.1.

In the case of file transfer, it is appropriate for a data protection field containing a CRC (cyclic redundancy check) or checksum to be included in each data unit, with any data unit for which this is inconsistent being discarded and a retransmission requested. The protection field should be part of the data unit, and thus not examined by the network; the end-systems can choose to use a level of protection that is appropriate to the application.

For most synchronous flows, requesting a retransmission would introduce an unacceptable delay. In some cases FEC (forward error correction) can be used to reconstruct the original data, but to protect against burst errors it must be distributed over time, which, again, introduces a delay. Some coding technologies for audiovisual data can successfully decode data that includes bit errors. In the case of uncompressed audio or video, if there is a bit error in a sample then that sample (but not the rest of the data unit) should be discarded and reconstructed by interpolation from adjacent samples; errors in the least significant bits should be ignored, however, as the corrupted sample will still be a better approximation to the original than an interpolated value.

NOTE   This is consistent with TCP/IP, where TCP and UDP (layer 4) protect the payload but IP (layer 3) only protects the header. Ethernet violates this principle by including protection of the whole packet (not just the header) at layer 2, so a single bit error will cause loss of an entire packet.

6.4.3 Resilience

- The network should as far as possible continue to provide a service in the presence of equipment failures or deliberate attacks.

The network should be designed such that failures (both broken links and malfunctioning equipment) are detected and isolated as quickly as possible, and flows (both synchronous and asynchronous) automatically rerouted with the minimum of disruption. Flows and connections should be prioritised so that where a large number are involved the most urgent can be rerouted first. Such a mechanism would also allow emergency traffic to be given priority when a major incident causes the network to be overloaded.

To provide the highest availability where required (such as for flows which contribute to live broadcasts, or for safety-critical monitoring of transport systems or industrial plant), an application should be able to request the network to provide more than one route for a flow, and as far as possible to keep the routes separate so that any single failure will only affect one of them. The application should be able to use the different routes to provide an appropriate amount of resilience, for instance if there are two routes it should have the option of sending two copies of the data (one on each route), or sending two copies of the most important part, with the remainder being shared between the two routes, or sending all the data on one route and keeping the other in reserve.

- The network should be able to resist denial-of-service and similar attacks.

6.4.4 Accounting

- The network should provide the necessary mechanisms to support per-call billing, similar to those used in telephone networks, and support the use of monetary cost as a routing criterion.

This would enable new business models, for instance monetising content by making it available via premium-rate calls.

It would also allow users to choose whether to pay (or pay more) for a higher-quality stream, and provide a mechanism for ISPs to recoup the cost of increasing capacity.

6.4.5 Time-limited calls

- The network should allow resources to be reserved ahead of requirement, for instance to carry an outside broadcast of a concert or sporting event.

- The network should allow a call to be connected for a specified time, in which case the network may disconnect it at any time after the specified period has expired.

6.4.6 Prioritisation of calls

- The network should be able to distinguish a small number of different priorities of call, and be able to disconnect lower-priority calls when required to release capacity for higher-priority calls.

NOTE   Three priority levels are probably sufficient: emergency, normal, and low. Emergency level would be reserved for emergency services.

6.5 Net neutrality

- The service received by a data unit should be independent of its contents.

For synchronous flows, the service should be negotiated when the flow is set up. At any point in the network, all data units that are part of an asynchronous flow should receive a similar service.

If access to a resource is refused, an explicit reason should be conveyed to the requester.

6.6 Mobility

- The network should support seamless handover of mobile and multi-homed devices from one point of attachment to another, without interruption to any synchronous flows being transmitted or received.

- The network should support seamless handover from one device to another.

This is to support use cases such as where someone who has been listening to a radio programme on their phone arrives home and wants to continue listening on their hi-fi.

- Where equipment has multiple attachment points, the network should support load sharing between them, and automatic rerouting if one fails (see 6.4.3).

If they connect via different service providers, the user may need to be offered a choice (see 6.4.4).

6.7 Scalability

- Addressing should be extensible.

With worldwide coverage, and particularly in the context of the Internet of Things, the network will need to support many billions of attached devices, all of which will need to be individually addressable.

- The network should not require globally-significant addresses to be included with every data unit.

In a connection-oriented network a switch or router only needs local information to correctly forward each data unit; provided the call set-up procedures support extensible addresses, connection-oriented networks are inherently more scalable than connectionless networks, which require the address to be part of the encapsulation of the data unit.

Many applications will be confined to a small subnetwork (such as a home network) plus a small number of other devices (such as the users' mobile phones). These applications should be able to perform actions, such as discovering all the devices on the subnetwork, that would not scale to a larger network.

6.8 Network management

- Management of the network should require as little manual intervention as possible. Setting up of devices newly-connected to a network should be largely automatic.

Switches and other equipment in public networks may require more configuration than those in home and small office networks, but functions such as load-balancing should be largely automatic. Network equipment

should use information provided by applications, as well as monitoring queue levels etc, to identify and report where changes (for instance, installing additional capacity or reconfiguring connectivity) are required.

- Faults should be detected, and promptly reported, in a way that makes as clear as possible the nature and location of the fault.

An application requiring access to a particular resource should be able to discover whether the resource is available, and the location of the problem (e.g. failed link) if it is not. It should not simply broadcast requests on the network and wait to see whether there is a reply.

- Applications should be able to negotiate parameters such as bandwidth with the network; in the case of compressed media, the network should be able to request a lowering of the bandwidth if it becomes congested, or offer an increase in bandwidth when congestion eases.

This may be associated with economic incentives (see 6.4.4).

6.9 Energy efficiency

- Switching technologies should be designed to use as little energy as possible.

Communications equipment already accounts for a significant part of the world's energy use. With increasing numbers of connected devices, and increasing bandwidth required for traffic such as higher definition video, there is a need to improve the energy efficiency of networking equipment.

Connection-oriented switching uses less power than connectionless: see 5.2.4.

All-optical networks consume less power than networks in which switching is in the electronic domain. The network should be able to offer the use an optical connection for traffic that would benefit from it. (This might include large file transfers, such as downloading a film, and live TV transmissions in very-high-definition formats, or bundles of TV channels.)

7 Gap analysis

7.1 Introduction

The current Internet uses connectionless packet switching, as do most local area networks and much of the telecommunications infrastructure.

This clause 7 examines the extent to which connectionless packet switching, and Internet Protocol in particular, is unable to satisfy the requirements identified in clause 6.

7.2 Service experienced by flows

7.2.1 Delay and throughput

Where packet switching is truly connectionless, with the network not maintaining any "state" information as regards individual data flows, submission of each transmitted packet to the network is a separate event. The network cannot predict what traffic will be offered, and can therefore offer no guarantees regarding the time a packet will be held in queues or whether it will be dropped because a queue is full.

In a closed network it may be possible to ensure that the offered traffic is sufficiently within the capacity of the network that packets will not be dropped and an upper bound for network delays can be calculated. However, this relies on external controls on the traffic, and changes such as attachment of additional equipment may invalidate the calculations. It is not something that the network itself can guarantee.

In current networks, "deep packet inspection" is used to attempt to identify the level of service needed by each packet. However, this can only yield the network's estimate of the requirements as there is no communication of the application's actual requirements. It is also unable to prevent more high-priority traffic being offered than the network can carry.

When "streaming" content from, for example, a video-on-demand server to a user, the content can be transmitted at a rate that allows the receiving device to build up a substantial reserve of data, and thus tolerate long delays and interruptions to the data flow. It can also request retransmission of packets that have been dropped, and in most such applications the impact on video or audio quality of a small number of dropped packets is relatively minor.

In practice, there are problems with receiving UDP-encapsulated flows through firewalls and units that implement Network Address Translation, so such "streamed" content is more usually served as a series of downloads of short media files.

For applications where network delays impact directly on the user experience, such as telephony, a trade-off can be made between delay and lost packets. A packet that arrives after the time at which its content should have been played out is effectively lost; reducing the amount of buffering in the receiving device reduces the delay but also increases the number of lost packets. Thus a balance can be struck between delays and intelligibility. However, as the network becomes more congested a point is reached at which conversation becomes impossible.

For some applications, the inability of the network to provide low latency and guaranteed delivery is more serious. This includes contribution to live broadcasts (see Annex A) and any application in which the transmission is safety-critical. Many such applications cannot rely on the current Internet, but would be able to rely on a network that could offer the necessary guarantees.

7.2.2   Packet size

Maximum packet size (or Transmission Unit, MTU) is a property of the underlying network; on a standard Ethernet network it is typically 1500 octets including IP and (if present) SNAP headers, but if "jumbo" frames are used it is around 8000 octets. It should be at least 512 octets including the IP header for IPv4 networks, and 1280 for IPv6.

In IPv4 networks, packets that are larger than the MTU may be fragmented provided the "do not fragment" flag is not set, but this facility adds complexity and is not available for IPv6. A packet that is larger than the MTU and cannot be fragmented is discarded and an error message returned to the sender specifying the MTU. The sender must then repack the data into packets conforming to the new MTU and try again; the new packets may be rejected further along the path, in which case the process is repeated.

Because different packets may travel along different paths, it is quite possible that the path actually taken by a packet could have accommodated a larger packet size. It is also possible that a flow will be rerouted at any time to a path with a smaller MTU, causing an interruption to the flow while the new MTU size is established and the data repacked.

7.3   Addressing

Connectionless packet networks have to include addressing information in the header of each packet. The address also acts as a label, and its format is therefore tightly coupled to the switching technology. The address needs to be globally unique, whereas it is more efficient if labels have local scope because they can then be smaller and more easily processed, and do not restrict scalability of the system.

Most addressing of resources in the Internet uses domain names, whereas packets contain IPv4 or IPv6 addresses. A separate protocol such as DNS or SIP is used to discover the address to be used in packets. In the case of LAN technologies such as IEEE 802.3 and 802.11, a further protocol, ARP, is required to discover the MAC address, which is the label for layer 2 switches.

Devices such as personal computers and smart-phones, which already have MAC addresses, typically acquire an IP address via DHCP; there is no explicit mechanism for a device to signal it has finished using an

address, so the supply of addresses can be exhausted if there are a large number of devices used in scenarios such as hot-desking.

7.4 Security and accountability

7.4.1 Privacy, trust and traceability

Information such as the sender's address in a packet cannot be verified by the network. Authentication requires mechanisms such as public key encryption.

7.4.2 Resilience

Each packet is in theory routed independently, so in the event of a broken link or equipment failure along the path taken by packets in a flow, the packets should simply start taking a different route. In practice, routing information is cached in switches so that it does not need to be discovered afresh for every packet, and there are no formal protocols for updating this information when failures occur.

The network attempts to deliver all packets submitted to it to their destination address. It is possible for a large number of packets to be submitted from different sources for the same destination, either in a deliberate "denial of service" attack or because some external event causes many users to attempt to access a site at the same time. There is little that the system can do in these circumstances other than to drop most of the incoming packets.

7.4.3 Accounting

There is no mechanism to support functions such as per-call billing. Users can be charged according to the number of bytes or packets transferred, but in this case all packets are treated equally, and users may be charged for traffic they were not aware they were generating, such as when applications check for and download updates.

7.4.4 Time-limited calls

There is no mechanism to support this function.

7.4.5 Prioritisation of calls

There is no mechanism to support prioritisation of individual calls. Packets can be marked with an indication of the required priority, but there are no formal mechanisms to limit the number of packets submitted with the highest priority.

7.5 Net neutrality

In theory all packets should be treated equally. In practice many networks use methods such as Deep Packet Inspection to discriminate between packets, for instance to give those that would appear to benefit from lower latency priority over others, or to discard packets they disapprove of. However, there is no method for applications to signal their requirements to, or negotiate with, the network, nor are the assumptions made by the network well-documented.

7.6 Mobility

IP addresses identify an interface, and include the address of the subnetwork to which it is attached. Therefore, when a device moves from one network to another, or begins using a different interface which is connected to a different network, its IP address will change.

A number of additional protocols have been added to IP to provide a work-around for this problem. For details see Annex A of ISO/IEC TR 29181-4.

7.7 Scalability

Connectionless routing requires each packet to carry source and destination addresses in a fixed format, which in the case of IPv4 imposes an upper limit of about 4 billion on the total number of endpoints. IANA issued the last blocks of IPv4 addresses to regional registries at the beginning of February 2011, and in April 2011 the Asia Pacific regional registry began severely restricting allocations of its remaining IPv4 addresses.

7.8 Network management

Protocols such as DHCP enable the automatic setting-up of newly-connected devices, but DHCP servers, firewalls, etc, still need manual configuration for all but the simplest use cases.

Diagnosis of faults is often poor, for instance with applications simply reporting an inability to retrieve information from a website, giving no indication of where in the path from the PC to the LAN to the WAN to the server the fault might be, or even of whether the DNS lookup had succeeded.

Applications can measure changes in delays or packet loss, and adjust parameters accordingly, but there is no formal method for controlling transmission rates. If an application sending live media detects packet loss (which is probably caused by congestion), there is no incentive for it to reduce the data rate; the usual response is to add information which will allow the lost packets to be reconstructed (forward error correction), which increases the data rate. If a significant proportion of the traffic reacts in this way, it also increases the congestion.

7.9 Energy efficiency

Connectionless packet switching requires the header of every packet to be interpreted and the destination address and other information to be processed. Usually this requires searching content-addressable memory, whereas CO packets can be routed using a simple table look-up which uses much less power.

Packet switching relies on being able to store each packet in memory as it arrives, and then copy it to the output port when it reaches the head of the queue, so has to be done in the electronic domain because optical memory is not available. It is therefore unable to take advantage of the lower power consumption of all-optical networks, except for some trunk connections between routers.

8 Functional architecture for switching and routing

8.1 Model of the switching and routing process

This document assumes the following model of the process of switching and routing data in communications networks. It is intended to support both legacy network technologies and the new technologies currently being developed.
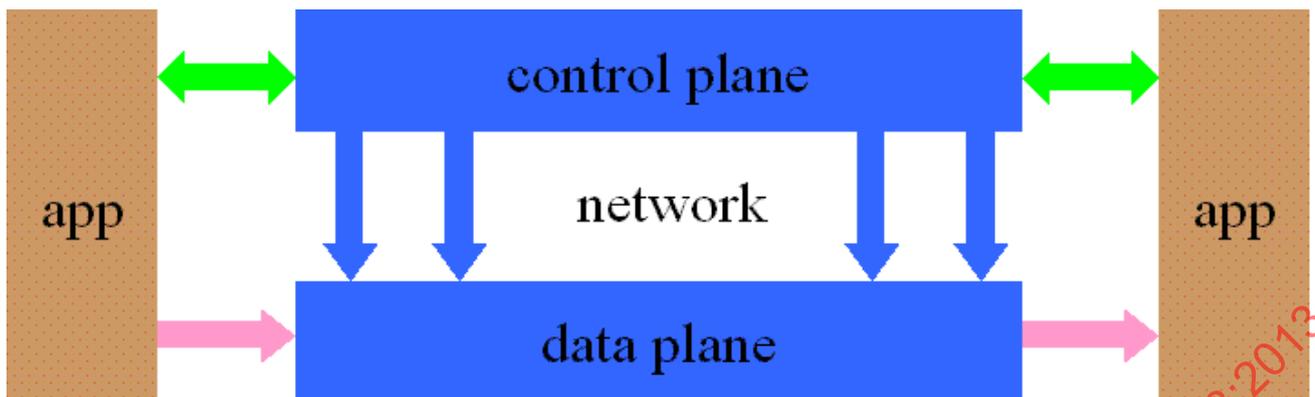
Figure 1 — Simplified model of communication

In Figure 1 two "application" entities are shown communicating across a network. Their interaction with each other and with the network is divided into two planes, "control" and "data".

The control plane carries messages which relate to the service which a flow will receive from the network. This includes establishing the labelling required in the data units to ensure they reach the required destination as well as negotiating QoS etc parameters. It also carries messages between the endpoints, carrying information such as higher-layer protocols to be used and media coding details. Examples of control plane protocols are ITU-T Q.931 [8] and Q.2931 [9] for connection-oriented networks, and ARP, DNS, and SIP for connectionless networks.

The data plane carries the application's data units, routed according to the flow to which they belong (see 6.2). In Figure 1 a single unidirectional flow is shown, but a single call can include multiple flows and each flow can be in either direction. How these flows are identified and related to the call is specific to the switching technology, as are any restrictions on the number of flows. In the case of Q.2931, there can only be one flow in each direction for a call, and flows are identified by a VPI/VCI value which is in general different on each link the flow passes over. In the case of SIP, flows are identified by sender and destination addresses and port numbers; this identification is not changed by the network except when passing through equipment that performs Network Address Translation.

In switching and routing equipment, data plane functions will usually be implemented in dedicated hardware, while control plane functions will usually be implemented in software running in a computer which can configure the data plane to route data units appropriately. Data plane functions should therefore be kept simple, while control plane functions should not be expected to operate at line speed.

In packet networks, control plane messages will usually be conveyed between adjacent pieces of equipment by the same physical links that carry data flows, distinguished from them by particular label values. For example, Q.2931 messages use VCI 5, and control messages on IP networks are distinguished either by being addressed to the routing engine or by specific protocol identification or port numbers. The call management process therefore has the status of an application, but one for which predefined flows are used. However, the model is also applicable to systems in which the control plane is completely separate from the data plane, for instance where the data plane uses wavelength division multiplexing (with each flow having its own lambda), or in cross-point routers for point-to-point digital (or even analogue) audio and video where the control messages are carried by asynchronous serial links or over Ethernet.

8.2   Structure of the network

A network is composed of FN-aware network elements (see 3.8) and "links", which convey data flows between FN-aware network elements. A link may be a dedicated point-to-point link, such as Ethernet over Cat5e cabling, or a leased line connection over a telecommunications network, or a lambda on a DWDM network; or it may be a shared-media link such as IEEE 802.11. It may also be a legacy subnetwork (switched or shared-media) that connects more than two ports, or a larger legacy network such as a switched telecommunications network or the current Internet.

The route followed by a data unit from the source to the destination may need to traverse more than one kind of network. For instance, audio being transmitted from a studio in one location to a studio in another location will probably travel over the local infrastructure in both studios and a wide area network that links them; these three networks may well all use different data-plane technologies.

8.3 Layers

The concept of "layers" has evolved somewhat over the decades since it was introduced with the OSI model [10]. The fundamental premise remains valid, that each layer in a network element communicates with its peer layer in another network element, using the service provided by the layer below, in order to provide a service to the layer above.

The architecture specified in this document covers switching and routing, which is layer 3 in the OSI model. However, the control and data planes are treated separately.

The service assumed from the layer below for the data plane is conveyance of data units between network elements. The service provided to the layer above is conveyance of data units between end-systems. We do not assume there will be an identifiable layer 3 protocol; the objects conveyed between network elements will usually consist of a data unit and some encapsulation, which will include framing and a means of identifying the flow to which the data unit belongs, and the switching process will usually involve replacing the encapsulation for the incoming link with one for the outgoing link.

The data unit should not be inspected or altered by the network as it is conveyed from one piece of end equipment to the other, and the encapsulation should not be visible to processes in the end equipment.

This document does not specify how the processes that produce and consume data units in end equipment should be structured, nor any structure to the data units themselves. For some applications it will be appropriate for the data units to be "raw" application data; for others it will be appropriate to include headers that are processed by a protocol stack in the end equipment, for instance to detect and correct errors (see 6.4.2). The coding of data units is considered in ISO/IEC TR 29181-6.

9 Requirements for data plane

9.1 General

Where a link uses a legacy technology, the encapsulation and details of the service are constrained by that technology.

- New technologies should keep the encapsulation to a minimum, so that they can convey small data units efficiently.

For instance, the time taken to collect sufficient audio samples to fill a large packet can add significantly to delays. Also, many protocols require frequent acknowledgements to be sent; these therefore form a significant proportion of Internet traffic, and are usually very small messages, the information content often being no more than a 2- or 4-byte sequence number.

In the case of synchronous flows, routing without using any information in the encapsulation of individual data units is possible. One example is ISDN, which carries single-byte data units that are routed according to their position in the frame. Such techniques may be useful for packet switching in optical networks.

- Wired links should not check for bit errors in individual data units. They should, however, include provision for detecting bit errors in headers, to reduce the probability of a data unit being misrouted.

Requirements differ between applications (see 6.4.2) and for many applications it is better to deliver a data unit with bit errors than to discard it. The error rate on each link should be monitored, and links with an unacceptable error rate taken out of service.

NOTE   A flag could be introduced that shows whether bit errors are tolerated (defaulting to true for synchronous flows, false for asynchronous flows) and errored data units for which it is false could then be discarded. However, packets with bit errors should be sufficiently rare that the gain is not worth the extra complexity. A simple mechanism for monitoring links to detect when bit errors are occurring, as in SDH, can be used to ensure that bit errors are, indeed, rare.

Similar considerations apply to wireless links; a data unit with a few bit errors should not be discarded, but a link that produces a significant number of errors should not be used.

9.2   Asynchronous service

- The data plane functionality should be conventional label switching augmented with the additional functionality described below.

This combines the advantages of connectionless and connection-oriented routing by providing all the functionality of a connectionless network with only a very small increase in the data plane complexity of connection-oriented switches.

Asynchronous paths used for this service are classified as "uplink" and "downlink". In a typical scenario, a number of clients share an uplink path to a server over which they send requests, and each has its own downlink path from the server over which it receives replies. Thus while downlink paths are normal one-to-one connections, uplink paths are many-to-one, and thus require additional encapsulation to identify the sender. However, unlike connectionless packets this identification does not need to be the sender's address in any particular address space, and will usually be meaningful only to the server, for instance a "handle" which identifies the session. It does not need to be processed by the network and should thus be defined by the application. For some servers a two- or four-byte prefix to the data unit will be sufficient, whereas in other cases an IP header (v4 or v6) might be used.

Data units on uplink and downlink paths have encapsulation (in addition to the encapsulation required by the link technology) which identifies the packet as being of one of four types: set-up, clear-down, initialisation, or data.

Uplink and downlink paths are connected using a protocol which is outlined below; a complete specification will be included in the FN standards. "Simple" asynchronous paths, which do not have the additional functionality of uplink and downlink paths may also be supported, connected using the protocols described in clause 10; these paths are one-to-one and may be implemented as carrying data units without any additional encapsulation or as downlink paths on which only data packets are allowed.

In a typical scenario, a TCP session between a client and a server would proceed as follows.

a)   Client sends a set-up packet to the control plane entity of the switch to which it is connected. The data unit is a TCP SYN datagram. The additional encapsulation includes the server's address (maybe in the form of a URL), the label to be used for packets from the switch to the client on the downlink path, and any authentication information the server may require.

b)   As the set-up packet propagates through the network, it sets up both uplink and downlink paths in the data planes of the switches.

c)   When the set-up packet reaches a switch that already has an uplink path to the server, the switch forwards it on that path; subsequent switches still set up the downlink path, but without requiring any action by the control plane (except maybe to provide a pool of free labels that can be used for downlink paths).

d)   When the server receives the set-up packet it allocates a "handle" for the session, processes the TCP datagram, and sends on the new downlink path an initialisation packet which includes the "handle" value as additional encapsulation and its reply to the TCP datagram as the data unit.

e)   Subsequent TCP datagrams are sent in data packets; those on the uplink path include the "handle" value.

f) At the end of the session the server sends a clear-down packet which includes a TCP FIN datagram. Each switch that forwards this packet also clears down the downlink path.

The additional functionality required of the data plane is thus (1) copying routing information from the set-up packet into a free location in its routing table (probably taken from a pool of free locations supplied by the control plane), (2) replacing the downlink path label in the set-up packet with one selecting the chosen location in its routing table, and (3) freeing up the location that is used to route a clear-down packet. It should also include flags in its routing table entries that identify uplink and downlink paths, and reject any packets that are not legal on the type of path on which they arrive. Provided the packet format is chosen to be such that the relevant fields are easy to locate, none of these functions should introduce significant extra complexity.

Where the server's address is an IP address, the control plane's task in the early stages is the same as in an IP router, i.e. to discover which output leads towards the required address. From (c) onwards, there is no involvement by the control plane and therefore no delay to the packets other than queuing delays. Therefore, the session will complete in no more time than it would take on an IP network.

If the client already has an uplink path to the server, it sends the set-up packet on that path instead of to the control plane. If there is no existing uplink path to the server from anywhere in the network, the server will receive the set-up packet as a control message.

Connection of the downlink path should be optional, signalled by a flag in the set-up packet. Uplink paths can then also be used to carry connectionless (e.g. UDP) datagrams.

The control plane does not keep any record of downlink paths, though it can search the routing table for them; this should only be necessary when a link goes down.

The control plane does, however, keep a record of uplink paths. A client may request clearing down of an uplink path, but usually the path will be retained in case a further session with the same server is required. Uplink paths that are not used should be pruned; the mechanism for this will be detailed in the FN standards.

10 Requirements for control plane

10.1 Control plane protocol

The control plane protocol consists of messages exchanged between adjacent network elements ("adjacent" in the sense of being connected to the same link) in order to manage communication between applications in end-systems. For most systems, these messages will be conveyed by the data plane service in the same way as any other application data. The protocol has elements of layers 4, 5, 6, and 7 (transport, session, presentation, and application), but the description in this clause 10 does not explicitly identify to which layer each part of the protocol belongs.

The services it provides include: requesting connection to partners which may be identified in a variety of ways (including by a service they provide); specifying requirements for QoS parameters; managing connections (including requesting variations in throughput, and disconnecting); and sending short messages. It also delivers information including: incoming connection requests and short messages; actual values of QoS parameters; and other status information such as congestion along the route of a connection, and link failures causing either disconnection or rerouting.

IEC 62379-5-2 [12] specifies a protocol which meets these requirements.

10.2 Message format

10.2.1 Encoding

- Messages should be easy for embedded code running on low-performance processors to interpret; the coding of information in the message should be reasonably efficient.

A tag-length-value (TLV) format would be appropriate. A text format which requires scanning to identify line endings and keywords would not.

Numbers should be coded in binary rather than text. Variable-length data such as text strings should be in variable-length fields (not fixed-length fields of the maximum size). However, it is not necessary to use data compression techniques that would add complexity.

10.2.2 Size

- Messages longer than the link MTU should be supported.

Some messages will need to contain a large amount of information, so could be several kilobytes in length. On the other hand, it is reasonable for the link MTU to be 1500 octets (as in IEEE 802.3) or less.

10.2.3 Extensibility

- A network element receiving a message which contains information it does not recognise should still be able to process the remainder of the message.

A TLV format allows the recipient to skip any information elements with tags it does not support. Any format in which it cannot determine the extent of a field it does not recognise would not allow it to process information in the message beyond that field.

10.3 Relationship with link partner

- On point-to-point links, when the link comes up the network elements at each end of the link should exchange information including their identities and capabilities. This information should be periodically refreshed (less frequently if the link provides an indication of link down such as "loss of carrier").

The identity of the neighbour helps with establishing the network topology.

Even if "link down" is signalled explicitly by the physical layer, occasional checks should be made that the neighbour has not changed or locked up or been rebooted.

On shared-media links, including those based on IEEE 802.11, a message enquiring whether any other FN-aware network elements are connected should be transmitted on "link up". The detail of the message and the associated protocol, also whether the message should be repeated periodically thereafter, depend on the technology used for the link.

10.4 Timeouts

- Timeouts should only be used for recovery from failures, not (for instance) for inferring that a requested resource is not available.

Many protocols on IP networks send out a message and then wait, in some cases for 20 or 30 seconds, for a reply before inferring that there is nothing on the network offering the requested service. The control plane should always know what is available in the local area, so should always be able to send a "not available" reply immediately. If a link on which a request has been sent goes down before the reply has been received, this can be reported immediately.

10.5 Acknowledgements

- The protocol should provide for positive acknowledgement of receipt of messages, and for recovery if such acknowledgement, or a subsequent reply to the message, is not received. An acknowledgement should not be required if a reply is sent almost immediately.